

Automatic Open Knowledge Acquisition via Long Short-Term Memory Networks with Feedback Negative Sampling

Byungsoo Kim

Hwanjo Yu

Gary Geunbae Lee

Department of Computer Science and Engineering
POSTECH (Pohang University of Science and Technology)
Pohang, Republic of Korea
{bsmail90, hwanjoyu, gblee}@postech.ac.kr

Abstract

Previous studies in Open Information Extraction (Open IE) are mainly based on extraction patterns. They manually define patterns or automatically learn them from a large corpus. However, these approaches are limited when grasping the context of a sentence, and they fail to capture implicit relations. In this paper, we address this problem with the following methods. First, we exploit long short-term memory (LSTM) networks to extract higher-level features along the shortest dependency paths, connecting head-words of relations and arguments. The path-level features from LSTM networks provide useful clues regarding contextual information and the validity of arguments. Second, we constructed samples to train LSTM networks without the need for manual labeling. In particular, feedback negative sampling picks highly negative samples among non-positive samples through a model trained with positive samples. The experimental results show that our approach produces more precise and abundant extractions than state-of-the-art open IE systems. To the best of our knowledge, this is the first work to apply deep learning to Open IE.

1 Introduction

Open Information Extraction (Open IE) is a task that involves taking sentences and extracting the arguments and the relations between them. Open IE systems extract this information in the form of a triple or n-tuple. Consider the following input sentence: ‘*Boeing announced the 747 ASB in 1986*’. An Open IE system will extract *<Boe-*

ing; announced; the 747 ASB> and *<Boeing; announced the 747 ASB; in 1986>*, or *<Boeing; announced; the 747 ASB; in 1986>*. Open IE has been successfully applied in many NLP tasks, such as question answering (Fader et al., 2014), knowledge base (KB) population (Soderland et al., 2013), and ontology extension (Moro and Navigli, 2013). The major difference between traditional IE and Open IE is domain dependency. Traditional IE requires a pre-defined set of relations, whereas Open IE (Banko et al., 2007) does not. Open IE represents relations with the words in a sentence. This new paradigm removes domain dependency, extending the relation set to whole word-sets. Thus, it is possible to run Open IE at the scale of the Web.

Previous Open IE systems adopt two main approaches. The first approach involves manually defining the extraction patterns to find the relationships between arguments. Reverb (Fader et al., 2011) showed that simple parts-of-speech (POS) patterns can cover the majority of relationships. Gamallo et al. (2012) and KRAKEN (Akbiik and Löser, 2012) manually define extraction rules in dependency parse trees. The second approach involves automatically learning a set of dependency-based extraction patterns from a large corpus. Methods adopting this second approach include WOE (Wu and Weld, 2010), OLLIE (Mausam et al., 2012), and ReNoun (Yahya et al., 2014).

Although previous Open IE systems have been used in many other studies, these systems only extract relations that are represented explicitly in a sentence. For example, previous systems find the (explicit) relation of ‘*capital*’ between ‘*Vilnius*’ and ‘*Lithuania*’ in the following sentences: ‘*The two countries were officially at war over Vilnius, the capital of Lithuania*’; ‘*The geographical midpoint of Europe is just north of Lithuania’s*

capital, Vilnius’; and ‘*Vilnius was the capital of Lithuania, the residence of the Grand Duke*’. The explicit relations accompany text snippets, which are strong clues regarding the relation (‘*Vilnius, the capital of Lithuania*’, ‘*Lithuania’s capital, Vilnius*’, and ‘*Vilnius was the capital of Lithuania*’). However, previous Open IE systems fail to find the relation when it is implicitly represented in a sentence, such as ‘*He returned to Lithuania and then lived in the capital, Vilnius, until his death*’. Unlike explicit relations, an implicit relation is not captured merely with textual patterns. Extracting these implicit relations involves a deeper understanding of the context of a sentence.

In this paper, we propose a novel Open IE system that automatically extracts features using long short-term memory (LSTM) networks. The bi-directional recurrent architecture with LSTM units automatically extracts higher-level features along the shortest dependency paths connecting headwords of relations and arguments. Because these paths contain only informative words that are relevant to finding the proper arguments of the relation, the extracted features can grasp contextual information without superfluous information. Because there are no prevalent datasets for training Open IE systems, we propose methods for constructing training samples. In particular, feedback negative sampling selects highly negative samples among non-positive samples, and decreases disagreements between positive and negative samples. The procedure for constructing the training set is fully automatic. It does not require any manual labeling. The experimental results show that our proposed system produces 1.62 to 4.32 times more correct extractions, including implicit relations, with higher precision than state-of-the-art Open IE systems.

The remainder of this paper is organized as follows. Section 2 describes the two types of relations that our system aims to extract. Section 3 defines Open IE as two tasks: argument detection and preposition classification. Section 4 describes the procedure for automatically constructing the training set. Sections 5 and 6 provide detailed explanations of the neural network architectures for argument detection and preposition classification, respectively. Section 7 describes how triples are extracted from the outputs in argument detection and preposition classification. Section 8 describes experimental settings and shows evalua-

tion results. Finally, Section 9 concludes our work.

2 Types of Relation

The first type of relation is a verb-mediated relation. A relation of this type is a verb phrase. It often forms an n-ary relation. Consider as an example: ‘*Boeing announced the 747 ASB in 1986*’. The relation ‘*announced*’ has 3 arguments: ‘*Boeing*’, ‘*the 747 ASB*’, and ‘*1986*’. This n-ary relation is represented as an n-tuple: $\langle \text{Boeing}; \text{announced}; \text{the 747 ASB}; \text{in 1986} \rangle$. However, because a binary relation is a core concept of the semantic web and ontological KB, the n-ary relation must be converted to binary relations. This conversion involves handling the problem of incomplete relations. In the above example, by merely spanning all pairs of arguments, the triples are $\langle \text{Boeing}; \text{announced}; \text{the 747 ASB} \rangle$, $\langle \text{Boeing}; \text{announced in}; \text{1986} \rangle$, and $\langle \text{the 747 ASB}; \text{announced in}; \text{1986} \rangle$. However, the relation $\langle \text{Boeing}; \text{announced in}; \text{1986} \rangle$ omits critical information—namely, ‘*the 747 ASB*’—and fails to find the complete relation between ‘*Boeing*’ and ‘*1986*’. The appropriate triple, without loss of information, is $\langle \text{Boeing}; \text{announced the 747 ASB in}; \text{1986} \rangle$. Because ‘*the 747 ASB*’ is a patient of ‘*announce*’ in the case of $\langle \text{the 747 ASB}; \text{announced in}; \text{1986} \rangle$, the appropriate triple is $\langle \text{the 747 ASB}; \text{be announced in}; \text{1986} \rangle$. Note that we do not restore the complete passive form (‘*was announced in*’). Rather, ‘*be announced in*’ is sufficient for indicating the passive form and for downward application.

Another type of relation is a noun-mediated relation. A relation of this type is a noun phrase. As described in Yahya et al. (2014), a noun-mediated relation is an attribute of an argument. Consider as an example: ‘*He sat on the board of Meadows Bank, an independent bank in Nevada*’. A triple with a noun-mediated relation is $\langle \text{Meadows Bank}; \text{an independent bank in}; \text{Nevada} \rangle$. In this triple, ‘*Nevada*’ is the target of an attribute, ‘*an independent bank*’, and ‘*Meadows Bank*’ is the value of the attribute. We add ‘*be*’ to the relation phrase in order to specify its meaning as an attribute, resulting in $\langle \text{Meadows Bank}; \text{be an independent bank in}; \text{Nevada} \rangle$. Unlike verb-mediated relations, the conversion from a noun-mediated n-ary relation to binary relations merely involves spanning all pairs of arguments.

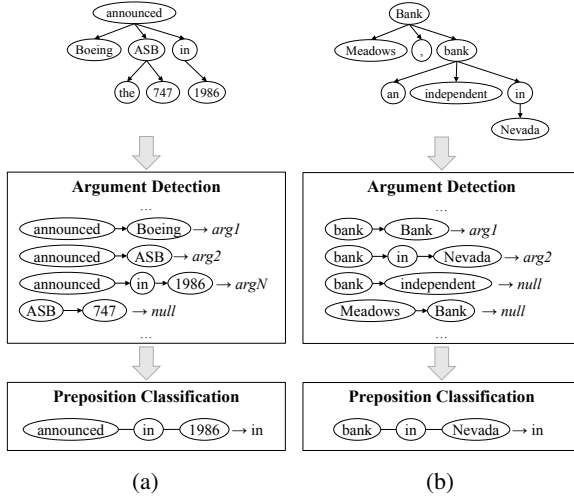


Figure 1: Argument detection and preposition classification, given the following sentences: (a) ‘Boeing announced the 747 ASB in 1986’, and (b) ‘Meadows Bank, an independent bank in Nevada’. For simplicity, we do not specify the dependency relations.

3 Task Definition

We define Open IE as two tasks: argument detection, and preposition classification. Given a sentence, detecting the argument involves regarding a certain word (*rel*) as a headword of a relation and then classifying other words (*arg*) as to whether they are the proper headwords of the arguments for that relation. As an input, the classifier takes the shortest dependency path connecting *rel* to *arg*. We denote this path as $path(rel, arg)$. By considering the shortest dependency path connecting two words, we can concentrate on informative words that are useful for understanding the relation between the two words (Bunescu and Mooney, 2005). For example, in Figure 1(a), ‘Boeing’, ‘ASB’, ‘the’, and ‘747’ are irrelevant for determining whether ‘1986’ is a proper argument for ‘announced’. We define four classes for argument detection: $arg1$, $arg2$, $argN$, and $null$. For a verb-mediated relation, $arg1$ and $arg2$ are the agent and patient of a relation, respectively, and $argN$ denotes other arguments. For a noun-mediated relation, $arg1$ and $arg2$ are the value and target of a relation, respectively. We do not classify $argN$ in the case of a noun-mediated relation. Finally, $null$ denotes a term that is not an argument. In Figure 1(a), argument detection classifies $path(announced, Boeing)$, $path(announced, ASB)$, and $path(announced, 1986)$ as $arg1$, $arg2$, and $argN$, respectively. Other paths are classified as

$null$. If the argument detection classifies $path(rel, arg)$ with the verb *rel* as $argN$ or the noun *rel* as $arg2$, the preposition classification finds the appropriate preposition between *rel* and *arg*. In Figure 1(a), preposition classification selects ‘in’ as the appropriate preposition between ‘announced’ and ‘1986’.

4 Automatically Constructing the Training Set

4.1 Highly Precise Tuple Extraction

As Christensen et al. (2011) leveraged semantic role labeling (SRL) to find n-ary relations, we used SRL¹ to extract highly precise tuples with verb-mediated relations². We assign *rel* to *predicate*, and $arg1$, $arg2$, and $argN$ to the labeled word with the roles *A0*, *A1*, and *AM*, respectively. If the word is a preposition, we apply the assignment to its child, while retaining the lemma of the preposition for preposition classification. Consider the following example: ‘In addition to the French Open, Nadal won 10 other singles titles in 2005’. The SRL output is ‘predicate: win, A0: Nadal, A1: titles, AM-DIS: In, AM-TMP: in’, and our assignment extracts the tuple as ‘rel: win, $arg1$: Nadal, $arg2$: titles, $argN$ (in): addition, $argN$ (in): 2005’. To minimize tuple-extraction errors, we only extract tuples from the top 1M sentences with the highest SRL confidence scores.

We define ten dependency-based extraction patterns to extract highly precise tuples with noun-mediated relations (see Figure 2). The patterns are applied to subgraphs of a dependency parse tree. The circles and arrows in the patterns represent words and dependency relations in the subgraph, respectively. For the preposition classification, we retain a lemma of a word at a circle with *IN*. If there is no circle with *IN* in a pattern, we retain ‘of’. If a pattern is matched, we assign $arg1$, *rel*, and $arg2$ to the words in the circles with $arg1$, *rel*, and $arg2$, respectively. For example, from the dependency parse tree of the sentence, ‘The agency is located in Gaborone, capital of Botswana’, the seventh pattern in Figure 2 is matched and our assignment extracts the tuple as ‘rel: capital, $arg1$: Gaborone, $arg2$ (of): Botswana’. Like verb-mediated tuple extraction,

¹We used ClearNLP (www.clearnlp.com) for the natural language processing pipeline.

²We used the English Wikipedia corpus to construct the training set.

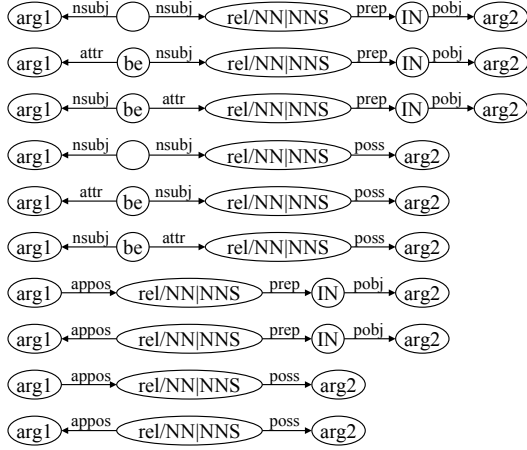


Figure 2: We restrict the POS tag of the relation to *NN* or *NNS*. Empty circles mean that we do not give any constraints. Circles with *IN* and *be* restrict the POS tag and lemma of the words to *IN* and *be*, respectively.

we only extract tuples from the top 1M sentences with the highest dependency-parsing confidence scores.

4.2 Training Set Augmentation

The goal of training set augmentation is to find sentences representing relations in highly precise tuples that SRL and the patterns failed to capture. Similar to OLLIE (Mausam et al., 2012) and Re-Noun (Yahya et al., 2014), this augmentation process is based on seed-based distant supervision: if arguments in a seed triple appear in a sentence, their relation is likely to appear in the sentence. The augmentation begins by converting the tuple to triples. For tuples with a verb-mediated relation, we convert each tuple to $\langle arg1; rel; arg2 \rangle$, $\langle arg1; rel; argN \rangle$, and $\langle arg2; rel; argN \rangle$. Tuples with a noun-mediated relation are converted to $\langle arg1; rel; arg2 \rangle$. Among the converted triples, we acquire 55K seeds satisfying the following constraints: (1) the arguments are proper nouns or cardinal numbers; (2) arguments with a proper noun are properly linked to entities in DBpedia (Auer et al., 2007); and (3) the lemma of a relation is not *be* or *do*. We use DBpedia Spotlight (Daiber et al., 2013) for entity linking. For each seed triple, we find sentences containing the same linked entities of arguments with a proper noun or the same surface forms of arguments with a cardinal number. Because the distant supervision hypothesis is often erroneous, we include the fol-

lowing constraints: (1) the sentence contains the lemma of a relation; (2) the headwords of relations and arguments are connected via a linear dependency path; and (3) triples with verb-mediated relations have a path length of less than seven. For example, we acquired the seed triple from the tuple, *rel: capital, arg1: Gaborone, arg2 (of): Botswana* with its arguments linked to DBpedia entities, *Gaborone* and *Botswana*. We retrieved the corpus and found *Now Prime Minister of Bechuanaland, Khama continued to push for Botswana's independence, from the newly established capital of Gaborone*. The augmentation produces 110K (sentence, seed triple) pairs that cannot be covered by highly precise tuples. We label $path(rel, arg1)$, $path(rel, arg2)$, and $path(rel, argN)$ from these pairs and highly precise tuples as $arg1$, $arg2$, and $argN$, respectively. These labeled paths comprise positive samples for argument detection. We also label $path(rel, argN)$ and $path(rel, arg2)$ with the noun *rel* as their prepositions to comprise samples for preposition classification.

4.3 Feedback Negative Sampling

Samples from the previous stages merely indicate which paths are $arg1$, $arg2$, and $argN$. They do not describe which paths are *null* (negative). One possible option for negative sampling is to regard non-positive paths as negative ones. However, this risks treating uncaptured positive paths as negative ones. For example, we found that $path(spoke, Moses)$ is an uncaptured positive path with the label $argN$ in the sentence: *Their presumption was rebuffed by God who affirmed Moses' uniqueness as the one with whom the LORD spoke face to face* (see Figure 3). Our strategy for addressing this problem begins from the observation that there are two features to a highly negative path: (1) it contains a positive path, or a positive path contains it; and (2) the more similar it is to the positive path, the more negative the path is. For example, in Figure 3, $path(rebuffed, their)$, $path(was, presumption)$, and $path(by, presumption)$ are highly negative paths. They contain $path(rebuffed, presumption)$, which is positive, and they have only one more node than the positive path. From this observation, we describe feedback negative sampling in Algorithm 1. The rationale behind this algorithm is as follows: because a model³ trained with positive samples assigns high confidence to

³We describe the details for this model in the next section.

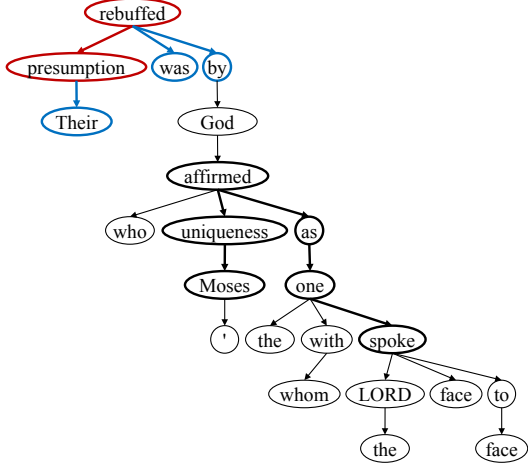


Figure 3: Feedback negative sampling acquires highly negative paths, such as $path(rebuffed, Their)$, $path(was, presumption)$, and $path(by, presumption)$, without mistakenly treating uncaptured positive paths, such as $path(spoke, Moses)$, as negative paths.

Algorithm 1: Feedback negative sampling

Input: NP = A set of non-positive paths

N = An empty set of negative samples

F = A model trained on positive samples

p = A prediction score threshold

foreach non-positive path $np \in NP$ **do**

$F(np)^i$ = Prediction score of np on the i -th class

$m = \arg\max_i F(np)^i$

if $F(np)^m > p$ **then**

$N = N \cup \{np\}$

return N

positive paths, it also assigns high confidence to non-positive paths with these features. For each non-positive path, we obtain clues (feedback) regarding these features. If the path has these features, the model assigns a high prediction score to a certain class. We regard the path as a negative sample when the score exceeds a certain threshold.

5 Argument Detection

Figure 4 describes the architecture for the neural network used for argument detection. At each time-step, the network acquires an input vector from a node in $path(rel, arg)$. The input vector is a concatenation of vectors from the following features: word, POS, dependency relation, and named entity.

Most of the deep learning applied NLP tasks

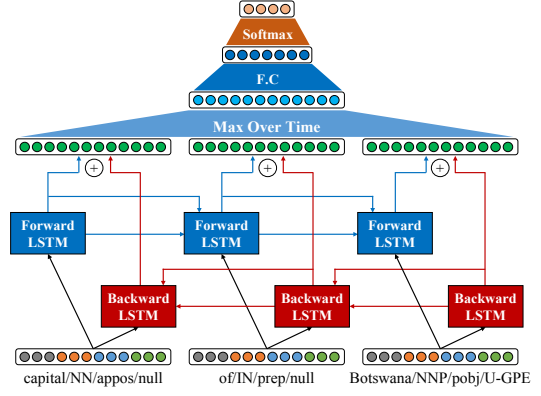


Figure 4: Neural network architecture for argument detection with input vectors from the $path(capital, Botswana)$ in the sentence, ‘The agency is located in Gaborone, capital of Botswana’.

leverage word embeddings trained with a large corpus in an unsupervised manner. The word embeddings capture the syntactic and semantic information of the words based on the context in the corpus. We pre-trained word embeddings from the English Wikipedia corpus with the skip-gram model in word2vec (Mikolov et al., 2013). In doing so, we acquired the word embedding matrix, $M_{word} \in \mathbb{R}^{dim_{word} \times |W|}$, where W is a set of words.

POS and dependency relations provide essential information regarding the syntactic structure of a sentence. However, there is no prevailing method for pre-training POS and dependency relation embeddings. In this work, we randomly initialized the POS and dependency relation embedding matrix, $M_{pos} \in \mathbb{R}^{dim_{pos} \times |P|}$ and $M_{dep} \in \mathbb{R}^{dim_{dep} \times |D|}$, where P and D are sets of POS tags and dependency labels, respectively. We then fine-tuned them in a supervised manner with back-propagation training.

Named entity recognition classifies each word into a pre-defined semantic category. We thus acquire the semantic types of words from the categories they belong to. Once again, the named entity embedding matrix, $M_{ne} \in \mathbb{R}^{dim_{ne} \times |N|}$ (where N is a set of named entity tags), is randomly initialized and updated through back-propagation training.

The word, POS tag, dependency label, and named entity tag of the t -th node in $path(rel, arg)$ are associated with a vector, $word_t \in \mathbb{R}^{dim_{word}}$, $pos_t \in \mathbb{R}^{dim_{pos}}$, $dep_t \in \mathbb{R}^{dim_{dep}}$, and $ne_t \in \mathbb{R}^{dim_{ne}}$ in the embedding matrix.

ces, respectively. We concatenate these vectors to produce a single input vector of the t -th node, $x_t = [word_t, post_t, dept_t, ne_t] \in \mathbb{R}^{dim_{word} + dim_{pos} + dim_{dep} + dim_{ne}}$.

A recurrent neural network (RNN) obtains the previous hidden state at each time-step, and creates and maintains the internal memory. By doing so, it can process arbitrary sequences of inputs. However, traditional RNNs have two well-known problems: vanishing and exploding gradients. If the input sequence is too long, the gradient can either decay or grow exponentially. An RNN with long short-term memory (LSTM) units was first introduced by Hochreiter and Schmidhuber (1997) in order to tackle this problem with an adaptive gating mechanism. Among the many LSTM variants, we selected LSTM with peephole connections in the spirit of Gers and Schmidhuber (2000). Furthermore, we use both the forward and backward directional recurrent LSTM layer (see Appendix A). This bi-directional architecture makes predictions based on information from both the past and the future. We obtain a bi-directional output vector $h_t \in \mathbb{R}^{dim_L}$ at each time-step from a vector sum of the forward (h_t^{fw}) and the backward (h_t^{bw}) LSTM layer output vectors.

$$h_t = h_t^{fw} + h_t^{bw} \quad (1)$$

We then convert an arbitrary number of bi-directional output vectors to a path-level feature vector h_{path} through a max-over-time operation (Collobert et al., 2011). This operation picks the salient features along the sequence of vectors to produce a single vector that is no longer related to the length of the sequence.

$$h_{path} = \max_t \{ (h_t)_i \} \quad (0 \leq i \leq dim_L) \quad (2)$$

Subsequently, a fully connected layer non-linearly transforms the path-level feature vector to learn more complex features. We select the hyperbolic tangent activation function to obtain a higher-level feature vector $h_{higher} \in \mathbb{R}^{dim_H}$.

$$h_{higher} = \tanh(M_{higher} \cdot h_{path}) \quad (3)$$

Finally, a softmax output layer projects h_{higher} into a vector with dimensions equivalent to the number of classes. The softmax operation is then applied to obtain a vector $h_{out} \in \mathbb{R}^4$ with its elements representing the conditional probability for each class.

$$h_{out} = \text{softmax}(M_{out} \cdot h_{higher}) \quad (4)$$

6 Preposition Classification

The neural network used for preposition classification is almost the same as the model used in the previous section. There are only two modifications to the preposition classification model. First, there is no penultimate fully connected layer in the model. We directly connect the max pooling layer to the softmax output layer. The second modification is to the number of output classes. The number of classes for preposition classification depends on the number of prepositions that appear in the positive samples. With 88 prepositions in the positive samples and one additional class for non-prepositions, the neural network model for preposition classification has $h_{out} \in \mathbb{R}^{89}$.

7 Triple Extraction

Triple extraction begins by aligning the prediction results as defined in the extraction template (see Table 1). This alignment produces incomplete triples of arguments and relations that are incomplete phrases. We span the dependents of aligned words, $arg1$, rel , $arg2$, and $argN$, to ensure that the triples contain sufficient information from the sentences.

Relation type	Triple template
Verb mediated	<[arg1]; [rel]; [arg2]>
	<[arg2]; be [rel] [prep]; [argN]>
	<[arg1]; [rel] [arg2] [prep]; [argN]>
Noun mediated	<[arg1]; be [rel] [prep]; [arg2]>

Table 1: Template for triple extraction.

Previous Open IE systems assign a score for each extracted triple. The score is used to indicate the degree of correctness, since extracted triples are not always correct. We define a scoring function as below.

$$\text{score}(t) = \text{dep}(s) \times \frac{\sum_{arg \in args} \text{prob}(arg)}{|args|} \quad (5)$$

where t is an extracted triple from a sentence s , $\text{dep}(s)$ is the dependency parsing confidence score of s , $args$ is a set of arguments in t , and $\text{prob}(arg)$ is the conditional probability of arg from the softmax output. Since errors in $\text{path}(rel, arg)$ are propagated to the final extraction, our scoring function is a mean of the conditional probabilities for arguments weighted by the dependency parsing confidence score of a sentence.

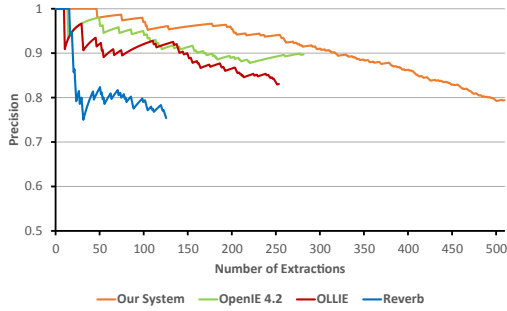


Figure 5: Our system produces more precise and abundant extractions than other state-of-the-art Open IE systems.

8 Experiments

8.1 Evaluation Settings

We crawled news articles on the Web and randomly sampled 100 sentences for evaluation. Because Open IE extracts totally new relations from the sentences, there is no ground-truth set of extractions. For this reason, our natural choice for a performance metric was to calculate the precision over the number of extractions. This is a common metric in previous Open IE studies. The extractions were manually annotated for correctness and sorted according to their score, in descending order. We set our system to output extractions with scores over 0.75, in order to clarify our evaluation results.

8.2 Comparison with State-of-the-Art Open IE Systems

We compared our system with three widely used Open IE systems: Open IE 4.2⁴, OLLIE, and Reverb. Unlike Open IE 4.2 and OLLIE, our system does not determine whether the extractions are factual. Thus, we considered all extractions from Open IE 4.2 and OLLIE in the comparison without distinguishing the factuality of the extractions. Because there is no way to convert unary relations to binary relations, we discarded unary relations from Open IE 4.2. Our proposed system produced more extractions than the other Open IE systems, and it achieved the highest precision in all areas regarding the number of extractions (see Figure 5). Specifically, the proposed system produced 1.62, 1.94, and 4.32 times more correct extractions than Open IE 4.2, OLLIE, and Reverb, respectively.

In addition to outperforming previous Open IE systems in terms of both precision and the to-

tal number of extractions, our system extracted implicit relations (see Appendix C). Extracting the implicit relations requires analyzing the context of the sentences, rather than merely setting boundaries to split the relations and arguments in sentences. Despite the relatively small proportion of implicit relations among correct extractions (3.8%), they were indeed worth extracting, because they contributed to more abundant extractions. We compared our system to a model trained with samples without the augmented training set and found that these extractions were made from properly learning the relations from the augmented training set. All Open IE systems, apart from the proposed system, failed to extract implicit relations. Open IE 4.2 heuristically converted SRL outputs to produce most of its extractions. Because of its high reliance on SRL, it missed the implicit relations that SRL failed to capture. In a manner similar to the augmented training set, OLLIE automatically constructed training samples with seed-based distant supervision. However, OLLIE converted dependency paths connecting headwords of relations and arguments into pattern templates. Consequently, OLLIE failed to extract complex features from sentences. Reverb assumed that arguments and their relations appear consecutively in a sentence. Although this assumption is often correct, it is unsuitable when extracting implicit relations.

8.3 Comparison with Different System Settings

Next, we analyzed how our system benefits from bi-directional LSTM networks (Figure 6(a)). We compared two sets of extractions: extractions from a model trained with samples from highly precise tuples (*Without Augmentation*), and extractions from a method using highly precise tuple extraction (*Highly Precise Tuples*). The former set contained 1.25 times more correct extractions than the latter set. Moreover, when quality extractions were considered, the first set contained extractions that were more precise.

We analyzed the effect of augmenting the training set by comparing two models: a model trained with the augmented samples (*With Augmentation*), and samples from the highly precise tuples (*Without Augmentation*). Figure 6(a) shows that augmented training set contributed to the production of 1.12 times more correct extractions with a slight

⁴<https://github.com/knowitall/openie>

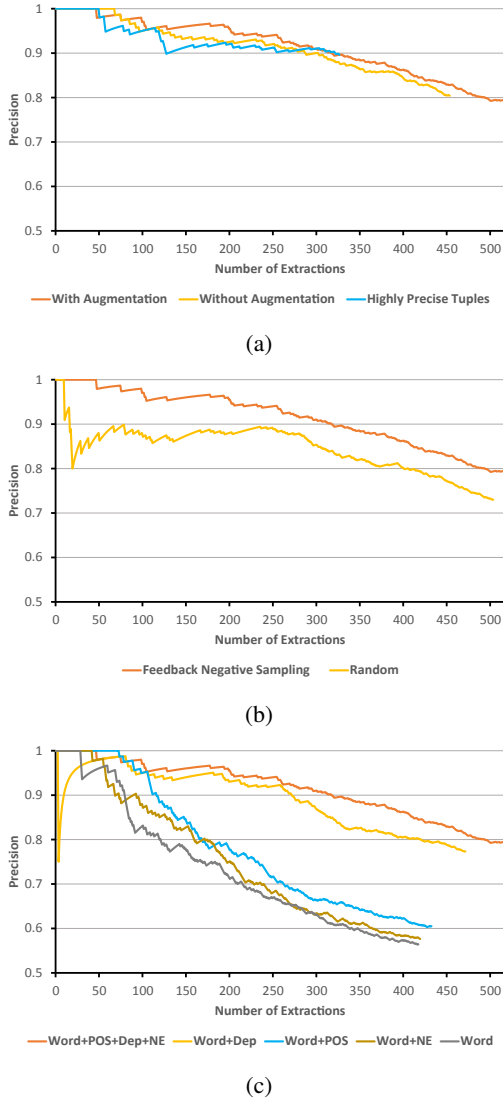


Figure 6: The best result is achieved with a model trained with positive samples from the augmented training set, negative samples from feedback negative sampling, and features from the word, POS, dependency relation, and named entity.

boost in precision. Furthermore, the model without augmentation produced no extractions with implicit relations.

We analyzed the quality of samples from the augmented training set. Because the samples were from (sentence, seed triple) pairs, we manually checked whether in each pair the seed triple represented a valid relation in the sentence. Among the 200 randomly sampled pairs, 83.5% were valid relations. Among pairs with invalid relations, 68% were due to a failure in the distant supervision assumption, 29% were due to errors in the seed triples, and 3% were due to entity linking errors.

We compared two negative sampling strategies:

feedback negative sampling (*Feedback Negative Sampling*), and random sampling of non-positive paths (*Random*) (see Figure 6(b)). Feedback negative sampling achieved higher precision overall. The loss of precision from random sampling of non-positive paths was due to disagreements between the positive and negative samples.

We also analyzed how each input feature contributed to the extraction performance (see Figure 6(c)). We set a baseline model with only the word feature (*Word*) as the input. We then added the POS (*Word+POS*), dependency relation (*Word+Dep*), and named entity (*Word+NE*) features one-by-one. Higher precision was achieved when the features were combined, compared to when only the word feature was used. Notably, the dependency relation feature boosted the precision considerably. By combining all four features (*Word+POS+Dep+NE*), the precision further increased, with the added advantage of expanding the total number of correct extractions.

8.4 Extraction Error Analysis

We analyzed incorrect extractions and investigated the source of the errors. According to our analysis, 20% of the errors were due to incorrect dependency parsing. Because the proposed system acquires a dependency path as an input, errors in dependency parsing were propagated throughout our system. Among the incorrect extractions without dependency parsing errors, 98% of the errors were from argument detection, and 4% were from preposition classification.

9 Conclusion

Our novel Open IE system with LSTM networks produced more precise and abundant extractions than state-of-the-art Open IE systems. In particular, the proposed system extracted implicit relations, unlike other Open IE systems. The advantages to the proposal stem from two contributions: a bi-directional recurrent architecture with LSTM units, enabling the extraction of higher-level features containing the contextual information in a sentence; and feedback negative sampling, which reduces the disagreements between positive and negative samples. To the best of our knowledge, this is the first work to apply deep learning to Open IE.

References

- Alan Akbik and Alexander Löser. 2012. Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, Montréal, Canada, June. Association for Computational Linguistics.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07*, pages 722–735, Berlin, Heidelberg. Springer-Verlag.
- Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07*.
- Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.
- Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the Sixth International Conference on Knowledge Capture, K-CAP '11*, pages 113–120, New York, NY, USA. ACM.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, November.
- Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems, I-SEMANTICS '13*, pages 121–124, New York, NY, USA. ACM.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.
- Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 1156–1165, New York, NY, USA. ACM.
- Pablo Gamallo, Marcos Garcia, and Santiago Fernández-Lanza. 2012. Dependency-based open information extraction. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pages 10–18, Avignon, France, April. Association for Computational Linguistics.
- F. A. Gers and J. Schmidhuber. 2000. Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000. IJCNN 2000*, volume 3, pages 189–194 vol.3.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *The Journal of Neural Computation*, 9(8):1735–1780, November.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea, July. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.
- Andrea Moro and Roberto Navigli. 2013. Integrating syntactic and semantic analysis into the open information extraction paradigm. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI'13*.
- Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S. Weld. 2013. Open information extraction to KBP relations in 3 hours. In *Text Analysis Conference, TAC'13*.
- Fei Wu and Daniel S. Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden, July. Association for Computational Linguistics.
- Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Lisbon, Portugal, September. Association for Computational Linguistics.

Mohamed Yahya, Steven Whang, Rahul Gupta, and Alon Halevy. 2014. Renoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 325–335, Doha, Qatar, October. Association for Computational Linguistics.

A Bi-Directional Recurrent Layer with LSTM Units

We begin from the forward-directional recurrent layer with LSTM units that receive input sequences from beginning to end (Equations 6-11).

$$f_t^{fw} = \sigma(W_f^{fw} \cdot x_t + U_f^{fw} \cdot h_{t-1} + V_f^{fw} \cdot c_{t-1}^{fw} + b_f^{fw}) \quad (6)$$

$$i_t^{fw} = \sigma(W_i^{fw} \cdot x_t + U_i^{fw} \cdot h_{t-1} + V_i^{fw} \cdot c_{t-1}^{fw} + b_i^{fw}) \quad (7)$$

$$g_t^{fw} = \tanh(W_g^{fw} \cdot x_t + U_g^{fw} \cdot h_{t-1} + b_g^{fw}) \quad (8)$$

$$c_t^{fw} = i_t^{fw} \otimes g_t^{fw} + f_t^{fw} \otimes c_{t-1}^{fw} \quad (9)$$

$$o_t^{fw} = \sigma(W_o^{fw} \cdot x_t + U_o^{fw} \cdot h_{t-1} + V_o^{fw} \cdot c_t^{fw} + b_o^{fw}) \quad (10)$$

$$h_t^{fw} = o_t^{fw} \otimes \tanh(c_t^{fw}) \quad (11)$$

There are four components in the LSTM unit: a forget gate f_t^{fw} , an input gate i_t^{fw} , a candidate memory content g_t^{fw} , and an output gate o_t^{fw} . The forget and input gate receive the current input x_t , the previous output h_{t-1} , and the previous memory content c_{t-1}^{fw} . These are then multiplied with the matrices W^{fw} , U^{fw} , and V^{fw} , respectively. Then, the multiplied values are summed with a bias b^{fw} , and the result is non-linearly transformed through the sigmoid function σ (Equations 6-7). The candidate memory content receives the current input and the previous output, which are multiplied with the matrices W^{fw} and U^{fw} , respectively. Then, the multiplied values are summed with a bias b^{fw} , and the result is non-linearly transformed through the hyperbolic tangent function \tanh (Equation 8). The output gate also receives the current input and the previous output, but it considers the current memory content, rather than the previous memory content (Equation 10). The current memory content is a combination of candidate memory content and previous memory content, weighted by the values of the input gate and the forget gate, respectively (Equation 9). Finally, the current output is a normalized current

memory content through the hyperbolic tangent function, weighted by the value of the output gate (Equation 11).

A potential problem with the forward LSTM layer is that it only considers information from the past. It thus fails to capture information from the future. We address this problem using an additional backward LSTM layer that receives input sequences from the end to the beginning (Equations 12-17).

$$f_t^{bw} = \sigma(W_f^{bw} \cdot x_t + U_f^{bw} \cdot h_{t+1} + V_f^{bw} \cdot c_{t+1}^{bw} + b_f^{bw}) \quad (12)$$

$$i_t^{bw} = \sigma(W_i^{bw} \cdot x_t + U_i^{bw} \cdot h_{t+1} + V_i^{bw} \cdot c_{t+1}^{bw} + b_i^{bw}) \quad (13)$$

$$g_t^{bw} = \tanh(W_g^{bw} \cdot x_t + U_g^{bw} \cdot h_{t+1} + b_g^{bw}) \quad (14)$$

$$c_t^{bw} = i_t^{bw} \otimes g_t^{bw} + f_t^{bw} \otimes c_{t+1}^{bw} \quad (15)$$

$$o_t^{bw} = \sigma(W_o^{bw} \cdot x_t + U_o^{bw} \cdot h_{t+1} + V_o^{bw} \cdot c_t^{bw} + b_o^{bw}) \quad (16)$$

$$h_t^{bw} = o_t^{bw} \otimes \tanh(c_t^{bw}) \quad (17)$$

B Training Details

We set the prediction score threshold p to 0.9 during feedback negative sampling. Furthermore, we set dim_{word} to 300, and dim_{pos} , dim_{dep} , and dim_{ne} to 50. Moreover, dim_L was set to 450, which was equivalent to the dimensions of the input vector, and dim_H was set to 50. Much like the regularization method used in Xu et al. (2015), we assigned the input vector a dropout rate of 0.5. Because we apply a softmax operation for the final output, the natural choice for a training objective is cross-entropy.

$$J(\theta) = \sum_{t \in T} \log p(y^{(t)} | x^{(t)}, \theta) \quad (18)$$

In the above equation, T is a set of training samples, and $\theta = (M_{pos}, M_{dep}, M_{ne}, M_{LSTM}, M_{higher}, M_{out})$ represents the network parameters, where M_{LSTM} denotes the parameters in the LSTM units. We used the ADAM (Kingma and Ba, 2014) update rule to maximize the training objective through stochastic gradient descent over shuffled mini-batches. We set β_1 to 0.9, β_2 to 0.999, and ϵ to 1e-8 for the ADAM parameters.

C Extraction Examples

System	Extractions	Annotation	
The UK Foreign Affairs Committee called upon Prime Minister David Cameron to boycott the event.			
our system	<The UK Foreign Affairs Committee; called upon; Prime Minister David Cameron> <The UK Foreign Affairs Committee; to boycott; the event> <David Cameron; be Prime Minister of; UK> <Prime Minister; called upon; David Cameron>	explicit explicit implicit	correct correct correct incorrect
OpenIE 4.2	<The UK Foreign Affairs Committee; called; Prime Minister David Cameron> <The UK Foreign Affairs Committee; called Prime Minister David Cameron to; boycott the event> <The UK Foreign Affairs Committee; called to boycott; the event>	explicit explicit explicit	correct correct correct
OLLIE	<The UK Foreign Affairs Committee; called upon; Prime Minister David Cameron> <The UK Foreign Affairs Committee; to boycott; the event> <The UK Foreign Affairs Committee; called to boycott; the event>	explicit explicit explicit	correct correct correct
Reverb	<The UK Foreign Affairs Committee; called upon; Prime Minister David Cameron>	explicit	correct
Article 7 of the UAE's Provisional Constitution declares Islam the official state religion.			
our system	<Article 7 of the UAE's Provisional Constitution; declares; Islam the official state religion> <Islam; be the official state religion of; the UAE> <Islam; declares; the official state religion>	explicit implicit	correct correct incorrect
OpenIE 4.2	<Article 7 of the UAE's Provisional Constitution; declares; Islam the official state religion>	explicit	correct
OLLIE	No extractions found.		
Reverb	<Article 7 of the UAE's Provisional Constitution; declares; Islam>	explicit	correct
It is 243 mi southeast of the capital Kiev on the Dnieper River, in the south-central part of Ukraine.			
our system	<It; is in; the south central part of Ukraine> <Kiev; be the capital of; Ukraine> <It; be 243 mi southeast of the capital Kiev on; the Dnieper River> <It; is; 243 mi southeast of the capital Kiev on the Dnieper River> <It; be 243 mi southeast of on the Dnieper River of; the capital Kiev>	explicit implicit explicit explicit	correct correct correct correct incorrect
OpenIE 4.2	<It; is; 243 mi southeast of the capital Kiev on the Dnieper River, in the south-central part of Ukraine>	explicit	correct
OLLIE	<It; is 243 mi southeast of the capital in; the south-central part of Ukraine> <It; is 243 mi southeast of the capital on; the Dnieper River> <It; is 243 mi of; the capital> <It; is; 243 mi southeast of the capital> <south-central; be part of; Ukraine>	explicit explicit	correct correct incorrect incorrect incorrect
Reverb	<It; is; 243 mi>		incorrect
The gigantic 37m Merlion Statue, representing the mascot and national personification of Singapore, was prominently seen above the promenade.			
our system	<The gigantic 37 m Merlion Statue; be seen; prominently> <The gigantic 37 m Merlion Statue; be seen above; the promenade> <The gigantic 37 m Merlion Statue; representing; the mascot and national personification of Singapore> <The gigantic 37 m Merlion Statue; be seen; representing the mascot and national personification of Singapore> <The gigantic 37 m Merlion Statue; be the mascot and national personification of; Singapore>	explicit explicit explicit explicit implicit	correct correct correct correct correct
OpenIE 4.2	<The gigantic 37 m Merlion Statue; representing; the mascot and national personification of Singapore> <The gigantic 37 m Merlion Statue; was prominently seen above; the promenade>	explicit explicit	correct correct
OLLIE	<The gigantic 37 m Merlion Statue; was prominently seen above; the promenade>	explicit	correct
Reverb	<the mascot; was prominently seen above; the promenade>		incorrect
The school was officially founded on August 22, 2014 when the NSHE Board of Regents approved a two year budget.			
our system	<The school; be founded; officially> <The school; be founded on; August 22 2014> <the NSHE Board of Regents; approved; a two year budget> <the NSHE Board of Regents; approved a two year budget on; August 22 2014> <a two year budget; be approved on; August 22 2014> <Regents; approved; a two year budget> <Regents; approved a two year budget on; August 22 2014>	explicit explicit explicit implicit implicit	correct correct correct correct correct incorrect incorrect
OpenIE 4.2	<The school; was officially founded; when the NSHE Board of Regents approved a two year budget> <The school; was officially founded on; August 22> <the NSHE Board of Regents; approved; a two year budget>	explicit explicit explicit	correct correct correct
OLLIE	<the NSHE Board of Regents; approved; a two year budget> <The school; was officially founded; 2014> <The school; was officially founded 2014 when the NSHE Board of Regents approved a two year budget on; August 22> <The school; was officially founded 2014 when the NSHE Board of Regents approved a two year budget in; August 22>	explicit	correct incorrect incorrect incorrect
Reverb	<The school; was officially founded on; August 22 , 2014> <the NSHE Board of Regents; approved; a two year budget>	explicit explicit	correct correct

Table 2: Our system extracts implicit relations missed by Open IE 4.2, OLLIE, and Reverb.